

EXHIBIT E- Riot Games's Developer Portal Overview

E1

<https://support-developer.riotgames.com/hc/en-us/articles/22698431229203-Developer-Portal-Overview>

Developer Portal Overview

11 months ago Updated

Overview

This document will provide you with a basic understanding of the Riot Games Developer Portal. It is designed to help you begin exploring and developing your own tools and products to positively impact the experience for players of Riot's games. We can't wait to see what you make!

Getting Started

Before you can begin taking advantage of the Developer Portal, you must login with your Riot Games account. Once you do, a Developer Portal account is created for you! This action also generates a basic development API key that is associated with your account. We'll talk about how you can interact with Riot to enhance that key further on in this documentation.

This account gives you the option to register your product proposal with Riot's Developer Relations team. Once you've registered your product you can communicate with our team for approvals, increased access, and more.

Product Registration

If you're here, you're probably thinking about making a cool application or website for a Riot game. For simplicity, we'll broadly call those things "products". As mentioned above, once you've logged in to the Developer Portal, you'll be able to register your product proposal. Let's go over what that entails.

Application Process

First off, every product owner (developer) will be expected to have read and understood our policies. You can find them in the nav bar at the top of this page. Read them. Failure

to adhere to these policies may lead to punitive actions, so make sure you understand them before going too far forward.

Even if your product doesn't use the Riot Games API, you'll find it beneficial to register your product. Registration lets Riot know you're out there, doing great stuff. If we know you're out there, then we can keep you informed of opportunities that might be beneficial to you and your product(s). Also, you will have a pathway to let us know how Riot can better help you and other developers.

So, how do you get registered?

It's simple. Head back to the main page of the Developer Portal and click the Register Product button. From there you'll need to decide if this is for a larger scale product, or a personal project. From there we'll ask you to fill out an online form that gathers some key details about your product. Finally, we'll ask you to verify your product, just to make sure you're really the developer!

Once you've completed those steps, our Developer Relations team will review your proposal. If everything is in order, we'll approve your product, and you're on your way to unlocking additional rate limits for your API key, and building a relationship with the team to help improve your product.

If your product is approved, we expect you to keep it in compliance of all applicable rules, policies, and laws. It's on you to stay up to date on all of those. If you need help with that, leave us a message in the Developer Portal, or jump into the Developer Discord.

If your product proposal is rejected, we'll be sure to leave you a message in the Developer Portal. We want you to succeed, so feel free to work with us through that messaging system to address our concerns, and get your great idea into the hands of players.

When reviewing applications we take a look at a lot of factors to determine if it's something that will benefit the Riot Games ecosystem. Of note, we want to make sure

that the product doesn't violate any laws or existing policies. We'll also want to make sure that the product helps players in some measurable way. We want to see product that help players get better at our games, or track their growth. What we don't want to see are products that solve our games or make everything too simple. Also, we are looking for quality. If your website isn't complete, we're unlikely to approve your product. If you have questions, though, just send us a message at our Support Site..

About that messaging system...

Messages

We'll send approval and rejection notifications to you directly in your application messages on the Developer Portal. If we have questions about a pending application, we'll be sure that you have a way to respond to us on the Developer Portal.

If you have questions specific to your product registration, you can send a message on our Support Site to get some feedback. Using the Support Site is the surest and quickest way to get a response from the Developer Relations team. Further, if you have any questions about your live products or anything else in the ecosystem, you can submit a message and we'll respond.

A caveat: Don't use the messaging system irresponsibly. This isn't a channel for you to casually chat with Rioters. This is a business channel, and should be treated as such.

Web APIs

Your product might depend on the Riot Games API. If it does, you'll need to be sure you understand some basic concepts about your Riot Games API key.

API Keys

Anyone who signs into the developer portal will automatically be granted an API key that will be associated with their account. Your API key allows you to start using the API

immediately, and allows us to generate API usage metrics for your key. In all, we manage several types of API keys. See the differences below.

DEVELOPMENT API KEYS

The API key that was generated for you when you signed into the developer portal is a development API key. These interim API keys are temporarily granted for products that are not meant for public consumption but benefit from temporary access to the API. The purpose of a development API key is for you to tinker with the Riot Games API and potentially develop a prototype for a product that you can make available for the community to use. They also deactivate every 24 hours. You'll need to regularly reset yours to keep it live.

PERSONAL API KEYS

You may apply for a personal API key by registering your product. Personal API keys should be used for products that are intended for just the developer or a small private community. These products can be registered without the verification process, but won't be approved for rate limit increases. You may request access to the Standard APIs, but not the Tournaments API. Personal keys require a detailed description of the product.

Acceptable uses for a personal API key include:

bots for streaming sites, boards, voice com servers, etc.

to display your own personal stats for your personal website

personal projects to gather your own stats

personal research

projects meant for personal usage and not production

The rate limit for a personal keys is by design very limited:

20 requests every 1 second

100 requests every 2 minutes

Note that rate limits are enforced per region. For example, with the above rate limit, you could make 20 requests every 1 second to both the NA and EUW League of Legends endpoints simultaneously.

You may not run your application for public consumption using a personal key, regardless of how long the approval process for your production key takes. Note that public consumption includes open alpha/beta tests.

PRODUCTION API KEYS

If you have developed an application using the Riot Games API that you would like to open up to players, you should apply for a production API key. You may not maintain a public product with a development API key. Production API keys have a much higher rate limit suitable for sustaining a public product's traffic.

Production API keys should be used for products that are intended for large communities or the Internet as a whole. You may request access to the Standard APIs and the Tournaments API. Typically requires a working prototype before receiving an API key.

The starting rate limit of a production API key is much larger than the development key:

500 requests every 10 seconds

30,000 requests every 10 minutes

Remember that this rate limit is enforced per region.

To apply for a production key with an expanded rate limit, click "Register Project" on your dashboard. The process and length of time required to obtain an approved production key can vary depending on your project and the application's target region(s).

Note that the standard production rate limit will meet the needs of the large majority of developers, but it can be expanded if the developer is in good standing, has demonstrated a strong community benefit, and has steadily outgrown the standard production limit.

If you are working on multiple projects, you should register each one separately and each one needs to be individually approved for a separate production API key.

API KEY SECURITY

As a final note on API keys, it's important to mention that your key will likely end up revoked if it isn't properly secured. Securing your API key is a requirement to publishing a project as outlined in the General Policies. If we determine that your key is not secured appropriately, we'll take action to secure it for you. :)

Response Codes

The Riot Games API returns all data in valid JSON. A few programming languages include native support for JSON. For those that don't, you can find a suitable library at <https://json.org/>.

Note that our APIs return only non-empty values to save on bandwidth. Zero is considered an empty value, as well as empty strings, empty lists, and nulls. Any numeric field that isn't returned can be assumed to be 0 (or null as you prefer). Any list field that isn't returned can be assumed to be an empty list or null. Any String field that isn't returned can be assumed to be empty string or null.

2XX RESPONSE CODES

For 200 response codes, you can always expect the response body documented on the API reference page. Only 200 response codes are guaranteed to return a response body as JSON.

For non-200 response codes please be aware of the following:

A response body is not guaranteed to be returned.

If there is a response body, its not guaranteed to be JSON.

We currently return JSON with human readable debugging information, but the structure and content of this debugging information are subject to change. As an example...

```
{  
  "status": {  
    "message": "Unauthorized",  
    "status_code": 401,  
  }  
}
```

The contents of status, message, and status_code are not guaranteed to always exist or remain constant for a given response code. 4. Logic within your application should fail gracefully based the response code alone, and should not rely on the response body.

4XX ERROR CODES

The 4xx class of error codes is meant to indicate that the client failed to provide a valid request. Below are the most common 4xx class of error codes you might encounter when using the API.

400 (Bad Request) This error indicates that there is a syntax error in the request and the request has therefore been denied. The client should not continue to make similar requests without modifying the syntax or the requests being made.

Common Reasons

A provided parameter is in the wrong format (e.g., a string instead of an integer).

A provided parameter is invalid (e.g., beginTime and startTime specify a time range that is too large).

A required parameter was not provided.

401 (Unauthorized) This error indicates that the request being made did not contain the necessary authentication credentials (e.g., an API key) and therefore the client was denied access. The client should not continue to make similar requests without including an API key in the request.

Common Reasons

An API key has not been included in the request.

403 (Forbidden) This error indicates that the server understood the request but refuses to authorize it. There is no distinction made between an invalid path or invalid authorization credentials (e.g., an API key). The client should not continue to make similar requests.

Common Reasons

An invalid API key was provided with the API request.

A blacklisted API key was provided with the API request.

The API request was for an incorrect or unsupported path.

404 (Not Found) This error indicates that the server has not found a match for the API request being made. No indication is given whether the condition is temporary or permanent.

Common Reasons

The ID or name provided does not match any existing resource (e.g., there is no Summoner matching the specified ID).

There are no resources that match the parameters specified.

415 (Unsupported Media Type) This error indicates that the server is refusing to service the request because the body of the request is in a format that is not supported.

Common Reasons

The Content-Type header was not appropriately set.

429 (Rate Limit Exceeded) This error indicates that the application has exhausted its maximum number of allotted API calls allowed for a given duration. If the client receives a Rate Limit Exceeded response the client should process this response and halt future API calls for the duration, in seconds, indicated by the Retry-After header. Applications that are in violation of this policy may have their access disabled to preserve the integrity of the API. Please refer to our Rate Limiting documentation below for more information on determining if you have been rate limited, and how to avoid it.

Common Reasons

Unregulated API calls.

5XX ERROR CODES

The 5xx class or error codes indicates that the server is aware it has errored or is incapable of performing the request. Below are the most common 5xx class of error codes you might encounter when using the API.

500 (Internal Server Error) This error indicates an unexpected condition or exception which prevented the server from fulfilling an API request.

503 (Service Unavailable) This error indicates the server is currently unavailable to handle requests because of an unknown reason. The Service Unavailable response implies a temporary condition which will be alleviated after some delay.

Rate Limiting

In order to control the use of the Riot Games API, we set limits on how many times endpoints can be accessed within a given time period. These limits are put in place to minimize abuse, to maintain a high level of stability, and to protect the underlying systems that back the API from being overloaded. The underlying systems are the same systems that power our games, so if they are overloaded, player experience suffers, and our first priority is to protect that experience.

RATE LIMITING TYPES

There are three types of limits used in the API infrastructure - application, method and service rate limits.

Application Rate Limits

The first type of limit is enforced on a per API key basis and is called an application rate limit. App rate limits are enforced per region. Every call made to any Riot Games API endpoint in a given region counts against the app rate limit for that key in that region. For example, calls to the static data API do not count against the application rate limit.

Method Rate Limits

The second type of limit is enforced on a per endpoint (or "method") basis for a given API key and is called a method rate limit. Method rate limits are also enforced per region. Every call made to any Riot Games API endpoint in a given region counts against the method rate limit for the given method and API key in that region.

Service Rate Limits

The third type of limit is enforced on a per service basis and is called a service rate limit. Service rate limits are also enforced per region. Every call made to any endpoint for a given Riot Games API service in a given region counts against the service rate limit for that service in that region. When service rate limits apply, we will document them, including which endpoints are part of the rate limited service.

Do not confuse method rate limits for service rate limits. Method rate limits apply individually to each application. Service rate limits apply to the service, and are shared by all applications making calls to a service.

Other Limits

These limits enforced by the API infrastructure are not the only gateways to the data provided. Some of the underlying services for certain endpoints may also implement their own rate limits, independently of the API infrastructure. In these cases, you will get a 429 error response, but there will be no X-Rate-Limit-Type header included in the response. Only when the rate limiting is enforced by the API edge infrastructure will this header be included.

While it is our policy not to reveal the specifics of how our rate limiting works, you can assume for the purposes of your code that the bucket starts when you make your first API call.

Versioning

As of 2019, these versioning guidelines apply to all Riot Games APIs. However, this may change in the future. Be sure to check documentation for each game you are developing products for to be sure you have the latest information.

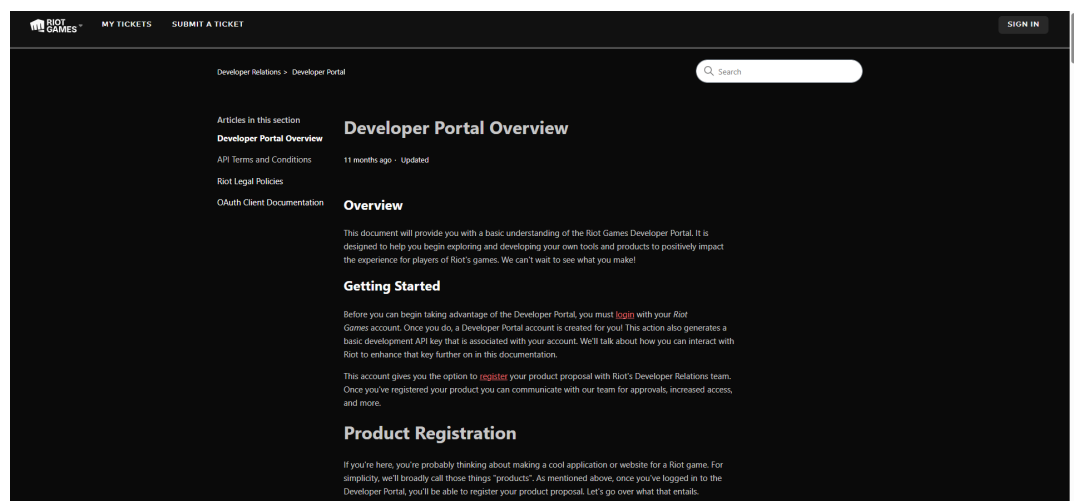
Currently, all League of Legends APIs are version 4. We no longer include a minor version in the API path.

Deprecation

As of 2019, these deprecation guidelines apply to all Riot Games APIs. However, this may change in the future. Be sure to check documentation for each game you are developing products for to be sure you have the latest information.

Whenever we deprecate, or make changes to, an API we aim to support both old and new versions of the API for 60 days. After that time the old version is deprecated. At times, circumstances require deprecation periods to be longer or shorter than 60 days. Deprecation periods will be communicated to developers through social media and the Developer Relations blog.

Developers should be careful when developing products with the Riot Games API. If an API is deprecated in favor of an updated version, the previous version will no longer be available. Any calls from a valid API key to a deprecated endpoint will result in an error code.



E2- https://developer.riotgames.com/docs/portal#_getting-started

Developer Portal

Overview

This document will provide you with a basic understanding of the Riot Games Developer Portal. It is designed to help you begin exploring and developing your own tools and products to positively impact the experience for players of Riot's games. We can't wait to see what you make!

Getting Started

Before you can begin taking advantage of the Developer Portal, you must login with your Riot Games account. Once you do, a Developer Portal account is created for you! This action also generates a basic development API key that is associated with your account. We'll talk about how you can interact with Riot to enhance that key further on in this documentation.

This account gives you the option to register your product proposal with Riot's Developer Relations team. Once you've registered your product you can communicate with our team for approvals, increased access, and more.

Product Registration

If you're here, you're probably thinking about making a cool application or website for a Riot game. For simplicity, we'll broadly call those things "products". As mentioned above, once you've logged in to the Developer Portal, you'll be able to register your product proposal. Let's go over what that entails.

Application Process

First off, every product owner (developer) will be expected to have read and understood our policies. You can find them in the nav bar at the top of this page. Read them. Failure to adhere to these policies may lead to punitive actions, so make sure you understand them before going too far forward.

Even if your product doesn't use the Riot Games API, you'll find it beneficial to register your product. Registration lets Riot know you're out there, doing great stuff. If we know you're out there, then we can keep you informed of opportunities that might be beneficial to you and your product(s). Also, you will have a pathway to let us know how Riot can better help you and other developers.

So, how do you get registered?

It's simple. Head back to the main page of the Developer Portal and click the Register Product button. From there you'll need to decide if this is for a larger scale product, or a personal project. From there we'll ask you to fill out an online form that gathers some key details about your product. Finally, we'll ask you to verify your product, just to make sure you're really the developer!

Once you've completed those steps, our Developer Relations team will review your proposal. If everything is in order, we'll approve your product, and you're on your way to unlocking additional rate limits for your API key, and building a relationship with the team to help improve your product.

If your product is approved, we expect you to keep it in compliance of all applicable rules, policies, and laws. It's on you to stay up to date on all of those. If you need help with that, leave us a message in the Developer Portal, or jump into the Developer Discord.

If your product proposal is rejected, we'll be sure to leave you a message in the Developer Portal. We want you to succeed, so feel free to work with us through that messaging system to address our concerns, and get your great idea into the hands of players.

When reviewing applications we take a look at a lot of factors to determine if it's something that will benefit the Riot Games ecosystem. Of note, we want to make sure that the product doesn't violate any laws or existing policies. We'll also want to make sure that the product helps players in some measurable way. We want to see product that help players get better at our games, or track their growth. What we don't want to see are products that solve our games or make everything too simple. Also, we are

looking for quality. If your website isn't complete, we're unlikely to approve your product. If you have questions, though, just reach out through the Developer Portal's messaging system.

About that messaging system...

Messages

We'll send approval and rejection notifications to you directly in your application messages on the Developer Portal. If we have questions about a pending application, we'll be sure that you have a way to respond to us on the Developer Portal.

If you have questions specific to your product registration, you can send a message on our Support Site to get some feedback. Using the Support Site is the surest and quickest way to get a response from the Developer Relations team. Further, if you have any questions about your live products or anything else in the ecosystem, you can submit a message and we'll respond. A caveat: Don't use the messaging system irresponsibly. This isn't a channel for you to casually chat with Rioters. This is a business channel, and should be treated as such.

Web APIs

Your product might depend on the Riot Games API. If it does, you'll need to be sure you understand some basic concepts about your Riot Games API key.

API Keys

Anyone who signs into the developer portal will automatically be granted an API key that will be associated with their account. Your API key allows you to start using the API immediately, and allows us to generate API usage metrics for your key. In all, we manage several types of API keys. See the differences below.

DEVELOPMENT API KEYS

The API key that was generated for you when you signed into the developer portal is a development API key. These interim API keys are temporarily granted for products that are not meant for public consumption but benefit from temporary access to the API. The

purpose of a development API key is for you to tinker with the Riot Games API and potentially develop a prototype for a product that you can make available for the community to use. They also deactivate every 24 hours. You'll need to regularly reset yours to keep it live.

PERSONAL API KEYS

You may apply for a personal API key by registering your product. Personal API keys should be used for products that are intended for just the developer or a small private community. These products can be registered without the verification process, but won't be approved for rate limit increases. You may request access to the Standard APIs, but not the Tournaments API. Personal keys require a detailed description of the product.

Acceptable uses for a personal API key include:

bots for streaming sites, boards, voice com servers, etc.

to display your own personal stats for your personal website

personal projects to gather your own stats

personal research

projects meant for personal usage and not production

The rate limit for a personal keys is by design very limited:

20 requests every 1 second

100 requests every 2 minutes

Note that rate limits are enforced per region. For example, with the above rate limit, you could make 20 requests every 1 second to both the NA and EUW League of Legends endpoints simultaneously.

You may not run your application for public consumption using a personal key, regardless of how long the approval process for your production key takes. Note that public consumption includes open alpha/beta tests.

PRODUCTION API KEYS

If you have developed an application using the Riot Games API that you would like to open up to players, you should apply for a production API key. You may not maintain a public product with a development API key. Production API keys have a much higher rate limit suitable for sustaining a public product's traffic.

Production API keys should be used for products that are intended for large communities or the Internet as a whole. You may request access to the Standard APIs and the Tournaments API. Typically requires a working prototype before receiving an API key.

The starting rate limit of a production API key is much larger than the development key:

500 requests every 10 seconds

30,000 requests every 10 minutes

Remember that this rate limit is enforced per region.

To apply for a production key with an expanded rate limit, click "Register Project" on your dashboard. The process and length of time required to obtain an approved production key can vary depending on your project and the application's target region(s).

Note that the standard production rate limit will meet the needs of the large majority of developers, but it can be expanded if the developer is in good standing, has demonstrated a strong community benefit, and has steadily outgrown the standard production limit.

If you are working on multiple projects, you should register each one separately and each one needs to be individually approved for a separate production API key.

API KEY SECURITY

As a final note on API keys, it's important to mention that your key will likely end up revoked if it isn't properly secured. Securing your API key is a requirement to publishing

a project as outlined in the General Policies. If we determine that your key is not secured appropriately, we'll take action to secure it for you. :)

Response Codes

The Riot Games API returns all data in valid JSON. A few programming languages include native support for JSON. For those that don't, you can find a suitable library at <https://www.json.org>.

Note that our APIs return only non-empty values to save on bandwidth. Zero is considered an empty value, as well as empty strings, empty lists, and nulls. Any numeric field that isn't returned can be assumed to be 0 (or null as you prefer). Any list field that isn't returned can be assumed to be an empty list or null. Any String field that isn't returned can be assumed to be empty string or null.

2XX RESPONSE CODES

For 200 response codes, you can always expect the response body documented on the API reference page. Only 200 response codes are guaranteed to return a response body as JSON.

For non-200 response codes please be aware of the following:

A response body is not guaranteed to be returned.

If there is a response body, its not guaranteed to be JSON.

We currently return JSON with human readable debugging information, but the structure and content of this debugging information are subject to change. As an example...

```
{  
  "status": {  
    "message": "Unauthorized",  
    "status_code": 401,
```

```
}  
  
}
```

The contents of status, message, and status_code are not guaranteed to always exist or remain constant for a given response code. 4. Logic within your application should fail gracefully based the response code alone, and should not rely on the response body.

4XX ERROR CODES

The 4xx class of error codes is meant to indicate that the client failed to provide a valid request. Below are the most common 4xx class of error codes you might encounter when using the API.

400 (Bad Request) This error indicates that there is a syntax error in the request and the request has therefore been denied. The client should not continue to make similar requests without modifying the syntax or the requests being made.

Common Reasons

A provided parameter is in the wrong format (e.g., a string instead of an integer).

A provided parameter is invalid (e.g., beginTime and startTime specify a time range that is too large).

A required parameter was not provided.

401 (Unauthorized) This error indicates that the request being made did not contain the necessary authentication credentials (e.g., an API key) and therefore the client was denied access. The client should not continue to make similar requests without including an API key in the request.

Common Reasons

An API key has not been included in the request.

403 (Forbidden) This error indicates that the server understood the request but refuses to authorize it. There is no distinction made between an invalid path or invalid

authorization credentials (e.g., an API key). The client should not continue to make similar requests.

Common Reasons

An invalid API key was provided with the API request.

A blacklisted API key was provided with the API request.

The API request was for an incorrect or unsupported path.

404 (Not Found) This error indicates that the server has not found a match for the API request being made. No indication is given whether the condition is temporary or permanent.

Common Reasons

The ID or name provided does not match any existing resource (e.g., there is no Summoner matching the specified ID).

There are no resources that match the parameters specified.

415 (Unsupported Media Type) This error indicates that the server is refusing to service the request because the body of the request is in a format that is not supported.

Common Reasons

The Content-Type header was not appropriately set.

429 (Rate Limit Exceeded) This error indicates that the application has exhausted its maximum number of allotted API calls allowed for a given duration. If the client receives a Rate Limit Exceeded response the client should process this response and halt future API calls for the duration, in seconds, indicated by the Retry-After header. Applications that are in violation of this policy may have their access disabled to preserve the integrity of the API. Please refer to our Rate Limiting documentation below for more information on determining if you have been rate limited, and how to avoid it.

Common Reasons

Unregulated API calls.

5XX ERROR CODES

The 5xx class of error codes indicates that the server is aware it has errored or is incapable of performing the request. Below are the most common 5xx class of error codes you might encounter when using the API.

500 (Internal Server Error) This error indicates an unexpected condition or exception which prevented the server from fulfilling an API request.

503 (Service Unavailable) This error indicates the server is currently unavailable to handle requests because of an unknown reason. The Service Unavailable response implies a temporary condition which will be alleviated after some delay.

Rate Limiting

In order to control the use of the Riot Games API, we set limits on how many times endpoints can be accessed within a given time period. These limits are put in place to minimize abuse, to maintain a high level of stability, and to protect the underlying systems that back the API from being overloaded. The underlying systems are the same systems that power our games, so if they are overloaded, player experience suffers, and our first priority is to protect that experience.

RATE LIMITING TYPES

There are three types of limits used in the API infrastructure - application, method and service rate limits.

Application Rate Limits

The first type of limit is enforced on a per API key basis and is called an application rate limit. App rate limits are enforced per region. Every call made to any Riot Games API endpoint in a given region counts against the app rate limit for that key in that region. For example, calls to the static data API do not count against the application rate limit.

Method Rate Limits

The second type of limit is enforced on a per endpoint (or "method") basis for a given API key and is called a method rate limit. Method rate limits are also enforced per region. Every call made to any Riot Games API endpoint in a given region counts against the method rate limit for the given method and API key in that region.

Service Rate Limits

The third type of limit is enforced on a per service basis and is called a service rate limit. Service rate limits are also enforced per region. Every call made to any endpoint for a given Riot Games API service in a given region counts against the service rate limit for that service in that region. When service rate limits apply, we will document them, including which endpoints are part of the rate limited service.

Do not confuse method rate limits for service rate limits. Method rate limits apply individually to each application. Service rate limits apply to the service, and are shared by all applications making calls to a service.

Other Limits

These limits enforced by the API infrastructure are not the only gateways to the data provided. Some of the underlying services for certain endpoints may also implement their own rate limits, independently of the API infrastructure. In these cases, you will get a 429 error response, but there will be no X-Rate-Limit-Type header included in the response. Only when the rate limiting is enforced by the API edge infrastructure will this header be included.

While it is our policy not to reveal the specifics of how our rate limiting works, you can assume for the purposes of your code that the bucket starts when you make your first API call.

Versioning

As of 2019, these versioning guidelines apply to all Riot Games APIs. However, this may change in the future. Be sure to check documentation for each game you are developing products for to be sure you have the latest information.

Currently, all League of Legends APIs are version 4. We no longer include a minor version in the API path.

Deprecation

As of 2019, these deprecation guidelines apply to all Riot Games APIs. However, this may change in the future. Be sure to check documentation for each game you are developing products for to be sure you have the latest information.

Whenever we deprecate, or make changes to, an API we aim to support both old and new versions of the API for 60 days. After that time the old version is deprecated. At times, circumstances require deprecation periods to be longer or shorter than 60 days. Deprecation periods will be communicated to developers through social media and the Developer Relations blog.

Developers should be careful when developing products with the Riot Games API. If an API is deprecated in favor of an updated version, the previous version will no longer be available. Any calls from a valid API key to a deprecated endpoint will result in an error code.

Declaration of Authenticity:

I, Marc Wolstenholme, declare under penalty of perjury that the statements made are true and accurate

Executed on April 28, 2025, in Coventry, England.

Respectfully submitted,

Signature: *M. WOLSTENHOLME*

Marc Wolstenholme

Plaintiff in Pro Per

5 Shetland Close

Coventry, England CV5 7LS

marc@mwwolf-fiction.co.uk